Amendments to the Claims:

This listing of claims will replace all prior versions, and listings, of claims in the application.

1. (Currently Amended) A byte code modification method, comprising:

   modifying a method's byte code instructions before said method's classfile has begun to be loaded for processing during runtime, said method comprising an entry point and an exit point, said modifying comprising inserting first and second additional byte code instructions into said method's byte code instructions, said first additional byte code instruction to invoke a first dispatch process as a consequence of said entry point being reached during runtime, said second additional byte code instruction to invoke a second dispatch process as a consequence of said exit point being reached during runtime; and,

   during runtime, invoking said first dispatch process from said first additional bytecode instruction and invoking said second dispatch process from said second additional byte code instruction, said first dispatch process ~~incleuding~~ including referring to a first dictionary to identify a first plug-in that performs a first output function, said second dispatch process including referring to a second dictionary to identify a second plug-in that performs a second output function.

2. (Original) The byte code modification method of claim 1 wherein said first output

function records a time of said entry point being reached.

3. (Original) The byte code modification method of claim 2 wherein said first output function records input parameters provided to said method.

4. (Original) The byte code modification method of claim l wherein said first output function records input parameters provided to said method.

5. (Original) The byte code modification method of claim l wherein said second output function records a time of said exit point being reached.

6. (Original) The byte code modification method of claim 5 wherein said second output function records output parameters provided by said method.

7. (Original) The byte code modification method of claim 1 wherein said second output function records output parameters provided by said method.

8. (Original) The byte code modification method of claim l wherein said first output function increments a counter maintained for said method.

9. (Original) The byte code modification method of claim 1 wherein said second output function increments a counter maintained for said method.

10. (Original) The byte code modification method of claim 1 further comprising

compiling source code prior to said modifying to produce said method's unmodified byte code instructions.

11. (Original) The byte code modification method of claim 1 wherein said byte code instructions are capable of being interpreted by a Java virtual machine.

12. (canceled).

13. (Previously Presented) The byte code modification method of claim 1 wherein said method's byte code instructions are capable of being interpreted by a Java virtual machine, and, said first additional byte code instruction is an invokestatic instruction.

14. (Previously Presented) The byte code modification method of claim 1 wherein said method's byte code instructions are capable of being interpreted by a Java virtual machine, and, said first additional byte code instruction is an invokevirtual instruction.

15. (Previously Presented) The byte code modification method of claim 1 wherein said method's byte code instructions are capable of being interpreted by a Java virtual machine, and, said first additional byte code instruction is an invokespecial instruction.

16. (canceled).

17. (canceled).

18. (Original) The byte code modification method of clam 1 wherein said modifying further comprises inserting a third additional byte code instruction, said third additional byte code instruction to cause a third output function to be executed for said method as a consequence of an error arising during execution of said method.

19. (canceled).

20. (Currently Amended) A byte code modification and distributed statistical recording method, comprising:

modifying a method's byte code instructions before said method's classfile has begun to be loaded for processing during runtime, said method comprising an entry point and an exit point, said modifying comprising inserting first and second additional byte code instructions into said method's byte code instructions, said first additional byte code instruction to invoke a first dispatch process as a consequence of said entry point being reached during runtime, said second additional byte code instruction to invoke a second dispatch process as a consequence of said exit point being reached during runtime;

during runtime, invoking said first dispatch process from said first additional bytecode instruction and invoking said second dispatch process from said second additional byte code instruction, said first dispatch process including referring to a first dictionary to identify a first plug-in that performs a first output function, said second dispatch process including referring to a second dictionary to identify a second plug-in that performs a second output function;

translating said information to a format employed within a distributed statistical records ("DSR") system.

21. (Original) The byte code modification and distributed statistical recording method of claim 20 wherein said first output function records a time of said entry point being reached.

22. (Original) The byte code modification and distributed statistical recording method of claim 21 wherein said first output function records input parameters provided to said method.

23. (Original) The byte code modification and distributed statistical recording method of claim 20 wherein said first output function records input parameters provided to said method.

24. (Original) The byte code modification and distributed statistical recording method of claim 20 wherein said second output function records a time of said exit point being reached.

25. (Original) The byte code modification and distributed statistical recording method of claim 24 wherein said second output function records output parameters provided by said method.

26. (Original) The byte code modification and distributed statistical recording method of claim 20 wherein said second output function records output parameters provided by said method.

27. (Original) The byte code modification and distributed statistical recording method of claim 20 wherein said first output function increments a counter maintained for said method.

28. (Original) The byte code modification and distributed statistical recording method of claim 20 wherein said second output function increments a counter maintained for said method.

29. (Original) The byte code modification and distributed statistical recording method of claim 20 further comprising compiling source code prior to said modifying to produce said method's unmodified byte code instructions.

30. (Original) The byte code modification and distributed statistical recording method of claim 20 wherein said byte code instructions are capable of being interpreted by a Java virtual machine.

31. (canceled).

32. (Previously Presented) The byte code modification and distributed statistical recording method of claim 20 wherein said method's byte code instructions are capable of being interpreted by a Java virtual machine, and, said first additional byte code instruction is an invokestatic instruction.

33. (Previously Presented) The byte code modification and distributed statistical recording method of claim 20 wherein said method's byte code instructions are capable of being interpreted by a Java virtual machine, and, said first additional byte code instruction is an invokevirtual instruction.

34. (Previously Presented) The byte code modification and distributed statistical recording method of claim 20 wherein said method's byte code instructions are capable of being interpreted by a Java virtual machine, and, said first additional byte code instruction is an invokespecial instruction.

35. (canceled).

36. (Original) The byte code modification and distributed statistical recording method of claim 35 wherein said handler method that performs said first output function and said handler method that perform said second output function are the same handler method.

37. (Original) The byte code modification and distributed statistical recording method of clam 20 wherein said modifying further comprises inserting a third additional byte code instruction, said third additional byte code instruction to cause a third output function to be executed for said method as a consequence of an error arising during execution of said method.

38. (Original) The byte code modification and distributed statistical recording method of claim 20 wherein said modifying further comprises inserting an additional byte code instruction for each of said method's exit points to cause a second output function to be executed for said method as a consequence of any of said method's exit points being reached.

39. (Currently Amended) A machine readable medium containing instructions which when executed cause a byte code modification method to be performed, the byte code modification method comprising:

modifying a method's byte code instructions<u> before said method's classfile has begun to be loaded for processing during runtime</u>, said method comprising an entry point and an exit point, said modifying comprising inserting first and second additional byte code instructions into said method's byte code instructions, said first additional byte code instruction to invoke a first dispatch process as a consequence of said entry point being reached during runtime, said second additional byte code instruction to invoke a second dispatch process as a consequence of said exit point being reached during runtime; and, during runtime, invoking said first dispatch process from said first additional bytecode instruction and invoking said second dispatch process from said second additional byte code instruction, said first dispatch process including referring to a first dictionary to identify a first plug-in that performs a first output function, said second dispatch process including referring to a second dictionary to identify a second plug-in that performs a second output function.

40. (Original) The machine readable medium of claim 39 wherein said first output function records a time of said entry point being reached.

41. (Original) The machine readable medium of claim 40 wherein said first output function records input parameters provided to said method.

42. (Original) The machine readable medium of claim 39 wherein said first output function records input parameters provided to said method.

43. (Original) The machine readable medium of claim 39 wherein said second output function records a time of said exit point being reached.

44. (Original) The machine readable medium of claim 43 wherein said second output function records output parameters provided by said method.

45. (Original) The machine readable medium of claim 39 wherein said second output function records output parameters provided by said method.

46. (Original) The machine readable medium of claim 39 wherein said first output function increments a counter maintained for said method.

47. (Original) The machine readable medium of claim 39 wherein said second output function increments a counter maintained for said method.

48. (Original) The machine readable medium of claim 39 wherein the byte code modification method further comprises compiling source code prior to said modifying to produce said method's unmodified byte code instructions.

49. (Original) The machine readable medium of claim 39 wherein said byte code instructions are capable of being interpreted by a Java virtual machine.

50. (canceled)

51. (Previously Presented) The machine readable medium of claim 39 wherein said method's byte code instructions are capable of being interpreted by a Java virtual machine, and, said first additional byte code instruction is an invokestatic instruction.

52. (Previously Presented) The machine readable medium of claim 39 wherein said method's byte code instructions are capable of being interpreted by a Java virtual machine, and, said first additional byte code instruction is an invokevirtual

instruction.

53. (Previously Presented) The machine readable medium of claim 39 wherein said method's byte code instructions are capable of being interpreted by a Java virtual machine, and, said first additional byte code instruction is an invokespecial instruction.

54. (canceled).

55. (canceled).

56. (canceled).

57. (canceled).